

文章编号: 1005-0523(2004)05-0037-04

S3C4510B 系统中对触摸屏控制的研究与实现

李中奇, 林知明, 韦宝泉

(华东交通大学 电气与电子工程学院, 江西 南昌 330013)

摘要: 触摸屏作为人机接口, 系统界面友好, 简单直观, 便于操作. 本文介绍了 S3C4510B 微处理器使用 ADS7846 控制芯片完成触摸屏模块的软硬件设计. 以及在嵌入式 Linux 操作系统中的软件驱动程序、应用程序开发.

关键词: S3C4510B; Linux; ADS7846; 触摸屏; 驱动程序

中图分类号: TP392

文献标识码: A

0 引言

1991 年 ARM 公司成立于英国剑桥, 主要出售芯片设计技术的授权. 目前, 采用 ARM 技术知识产权(IP)核的微处理器, 即我们通常所说的 ARM 微处理器, 已遍及工业控制、消费类电子产品、通信系统、网络系统、无线系统等各类产品市场. 在工业控制领域, 作为 32 的 RISC 架构, 基于 ARM 核的微控制器芯片不但占据了高端微控制器市场的大部分市场份额, 同时也逐渐向低端微控制器应用领域扩展. ARM 微控制器的低功耗、高性价比, 向传统的 8 位/16 位微控制器提出了挑战. S3C4510B 是 Samsung 公司生产的基于以太网应用系统的高性价比 16/32RISC 微控制器, 内含一个由 ARM 公司设计的 16/32ARM7TDMI RISC 处理器核. ARM7TDMI 为低功耗, 高性能的 16/32 核, 最适合用于对价格及功耗敏感的应用场合. 在这些工业控制场合, 触摸屏的使用非常广泛. 因此如何在 S3C4510B 系统中集成触摸屏模块, 以及在系统中实现其驱动程序都成为嵌入式系统设计者需要解决的问题.

1 系统硬件电路的实现

1.1 ADS7846 的工作原理

ADS7846 是 TI 公司出产的电阻式 4 线触摸屏的控制器, 是一种典型的 12 位取样的逐次近似寄存器(SAR)A/D 转换器. 它即能测量基本的触摸点位置, 还可进行触摸压力的测量. 芯片内部提供的 2.5 V 参考电压可用于辅助输入、电池监控器和温度测量. 其自动调节电功能可以保证很低的功率损耗, 对于要求低功耗的嵌入式系统电路非常适用.

电阻式触摸屏有两层电阻层: 一层为横向驱动, 另一层为纵向驱动. 工作时, ADS7846 在某个方向加一定的电压, 当笔接触到触摸屏时, 两电阻层在触点位置有一个接通, 导致另一个方向的层上电压产生变化. 通过 A/D 转换, 控制器就可以计算出触点在这条轴上的坐标值. ADS7846 中涉及到触摸屏工作的主要信号如下:

DIN	串行数据输入	DOUT	串行数据输出
DCLK	外部时钟输入	CS	芯片选通信号
X+	X 轴正向输入	Y+	Y 轴正向输入
X-	X 轴负向输入	Y-	Y 轴负向输入
PENIRQ	笔中断		

收稿日期: 2004-04-15

作者简介: 李中奇(1975-), 男, 黑龙江省哈尔滨人, 在读研究生.

其中 DIN 用于输入控制信号, DOUT 用于输出数据信号. 通常 ADS7846 工作在笔中断模式, ADS7846 检测到笔落下和抬起时, 发出中断信号, 微处理器在中断处理中通过向 DIN 写入控制字节可以设定 ADS7846 的工作模式、数据长度以及数据通道的选择, 并启动采样转换. ADS7846 在转换结束后, 从 DOUT 发出采样数值, 这时 CPU 可从于 DIN 相连的 I/O 管脚读进采样值, 进行处理.

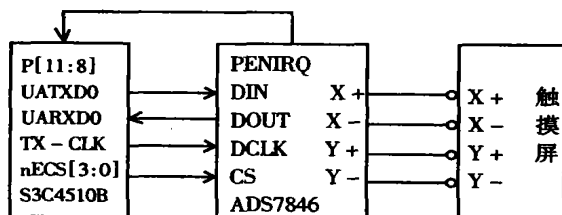
1.2 S3C4510B 与触摸屏接口电路的设计

除了 ARM7TDMI 核以外, S3C4510B 比较重要的片内外围功能包括: 2 个带缓冲描述符的 HDLC 通道, 2 个 UART 通道, 2 个 GDMA 通道, 2 个 32 位定时器, 18 个可编程的 I/O 口, 片内的逻辑控制电路包括: 中断控制器, DRAM/SDRAM 控制器, 系统管理器, ROM/SRAM 和 FLASH 控制器, 一个内部 32 位系统总线仲裁器, 一个外部存储控制器.

1) S3C4510B 的 UART 单元提供两个独立的异步串行 I/O 口 (Asynchronous Serial I/O, SIO), 每个通讯口均可工作在中断模式或 DMA 模式, 也即 UART 能产生内部中断请求或 DMA 请求在 CPU 和串行 I/O 口之间传送数据. 每一个异步串行通讯口都具有独立的波特率发生器、发送器、接收器和控制单元. 波特率发生器可由片内系统时钟 MCLK 驱动, 或由外部时钟 UCLK (Pin⁶⁴) 驱动; 发送器和接收器都具有独立的数据缓冲寄存器和数据移位器. 待发送的数据首先传送到发送缓冲寄存器, 然后拷贝到发送移位器, 并通过发送数据引脚 UATXD_n 发送出去. 接收数据首先从接收数据引脚 UARXD_n 移入移位器, 当接收到一个字节时就拷贝到接收缓冲寄存器.

2) SIO 的控制单元通过软件控制工作模式的选择、状态和中断产生.

当使用 UART 的发送中断功能时, 应在初始化 UART 之前写一个字节数据到 UART 的发送缓冲寄存器, 这样当发送缓冲寄存器空时就可以产生 UART 的发送中断. 在电路上, S3C4510B 的同步串行口可以和 ADS7846 无缝连接, 笔中断信号直接连到 S3C4510B 的通用 I/O 管脚.



2 嵌入式 Linux 系统下的驱动程序

Linux 是自由的多任务操作系统, 它需要 PC 桌面系统作为运行平台. 而本文所讨论的嵌入式 Linux 是指经过小型化裁剪、能够烧录入容量只有几百 KB 或几 MB 的闪存 (Flash Memory) 内, 不需要硬盘作为存储介质, 也不需要键盘、鼠标之类的外设, 适用于 8 位/16 位/32 位 MCU, 应用于各种特定嵌入式场合的专用 Linux 操作系统.

嵌入式 Linux 设备驱动程序中有一个很重要的数据结构 `file_operations {}`. 它是驱动程序与应用程序的接口, 使编写驱动程序的工作变得简单而规律. 在该触摸屏驱动程序中定义了一个数据结构为 `file_operations {}` 的变量 `touch_fops`, 并进行了如下的赋值:

```
static struct file_operations touch_fops =
{
    read: touch_read,
    write: touch_write,
    open: touch_open,
    release: touch_release,
    poll: touch_poll,
};
```

1) touch_open 函数

`touch_open()` 这个函数在 `file_operations {}` 中的原型是 `open()` 函数. 它的主要功能就是打开设备并初始化设备准备进行操作. 下面一段程序介绍了 `touch_open()` 函数的实现过程.

```
if ((ret = request_irq (IRQ_touchRX, touch_rx, 0, "touch_rx", dev_
idtouch)
{
    printk ("touch_rx_init: failed to register IRQ_touchRX n");
    free_irq (IRQ_touchRX, dev_idtouch);
    return ret;
}
```

在这个 if 语句中出现了 3 个函数. `printk()` 是 Linux 内核提供的函数, 功能近似标准 C 函数库中提供的 `printf()` 函数. 在 Linux 操作系统中, 因为驱动程序是在内核空间运行的, 所以必须使用内核提供的函数, `printf()` 不能在内核空间运行.

`Request_irq()` 是申请笔中断的函数, 其中参数 `IRQ_touchRX` 是所申请的中断号, `touch_rx` 是所安装的中断处理函数, 第三个参数是用于中断管理的一

些常量,这里的值为0,表示可以进行中断的共享,参数“touch_rx”是发送中断的设备名称,dev_idtouch是用来共享的中断号.如果成功申请中断的话则返回0给ret变量,当返回了一个非0的值给ret变量的时候,则说明有另外一个驱动程序已经占用了要申请的中断信号线.当申请中断失败后,就必须进行中断信号线的释放,使用的是free_irq()中断信号线释放函数.

2) touch_read 函数

touch_read()函数的原型是read()函数.它的作用就是从触摸屏设备中读取数据.当在file_operations{}结构中用NULL来表示此函数的话,则说明这个设备是不允许进行读操作的,如果对其进行调用的话,内核将会返回一个错误.下面来对这个touch_read()函数进行一下分析.

```
While(rx_user_count>0)
{
if((USTAT0&HCQ_RX_EMPTY_BIT)==0X40)
{
touch_rx_buf[rx_buf_count]=*URXBUF0;
rx_buf_count++;
}else {
interruptible_sleep_on_timeout(&rx_queue,TIMEOUT);
}
```

这个循环语句的作用是开辟一个缓冲区用来存放从触摸屏设备中传来的数据.其中interruptible_sleep_on_timeout()是延时函数,具体是用定时器来进行延时.延时的原因是由于触摸屏设备把数据传入缓冲区需要一定的时间,而CPU必须等数据进入了缓冲区后才能进行数据的读取.

3) touch_write 函数

touch_write()函数的原型是write()函数,它的主要作用是向触摸屏设备发送数据和命令.原理与touch_read()函数类似,只是数据传输的方向不同.

4) touch_release 函数

touch_release()函数的原型是release()函数.用来关闭触摸屏设备的.如果用NULL代替,则表示设备永远是关闭的.这个函数实现起来比较简单,先调用free_irq()释放触摸屏设备占用的中断控制线,接着使设备的引用计数为0.这样就完成了对触摸屏设备的关闭工作.

3 触摸屏的应用程序

Linux操作系统中应用程序工作在用户区.触摸屏应用程序通过已加载到内核模块中的驱动程序控制触摸屏.应用程序可以通过触摸屏实际使用情况来编写.在我们实际的测控系统中触摸屏做为输入设备,与液晶显示屏配合使用达到完成相应的按钮指令的功能.下面是测试触摸屏能否正常工作的应用程序.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <unistd.h>
FILE *fp;
int main()
{
char read_buf[2];
char write_buf[2]={GetX, GetY};
int fd, qr;
fd=open("/dev/touch", O_RDWR); //以可读写方式打开前面加
载的触摸屏驱动模块
if(fd<0)
{
printf("Error in open Device! \n"); //如打开失败,提示错误并
推出进程
exit(-1);
}
printf("open device ok! \n");
while(1)
{
write(fd, write_buf, 1);
qr=read(fd, read, 1);
if(qr>0)
{
GetLocation();
}
}
void GetLocation()
{
fp=fopen("/home/touch.txt", "w+");
write(fd, write_buf, 1);
fwrite(read(fd, read_buf, 1), 1, 1, fp);
write(fd, &write_buf[1], 1);
fwrite(read(fd, read_buf, 1), 1, 1, fp);
fclose(fp);
}
```

主程序通过 `fd = open("/dev/touch", O_RDWR)` 语句进行了一个 `open()` 函数的系统调用, 用来调用这个触摸屏驱动程序, 并以可读可写的方式来打开触摸屏, 把该 `open()` 系统调用的值返回给 `fd`, 作为判断打开该触摸屏是否成功. 接下来程序用 `while (1)` 来进行循环检测触摸屏是否有触发动作发生, 当有触发动作发生时调用按键处理程序来进行触点位置的读取.

按键处理程序用来得到触点的位置. 调用一个写数据的函数向触摸屏控制器发送命令, 然后触摸屏根据 CPU 所发送的命令确定返回的坐标数据. 这个程序中是先发送读取 X 坐标的命令, 然后发送读取 Y 坐标的命令, 最后得到触点的坐标值并打印出来.

4 结束语

本系统已成功运用于便携式电力机车受电弓检测仪. 为系统提供了便捷的人机接口, 友好的人机界面使测试系统更个性化, 测试员上手周期短, 劳动强度减少, 系统便于维护.

参考文献:

- [1] ALESSANDRO RUBINI. LINUX 设备驱动程序[M]. 北京: 中国电力出版社, 2000.
- [2] 马忠梅. ARM 嵌入式处理器结构与应用基础[M]. 北京: 航空航天大学出版社, 2002.
- [3] 杜春雷. ARM 体系结构与编程[M]. 北京: 清华大学出版社, 2002.

Study and Implementation of Control Touch Screen in S3C4510B System

LI Zhong-qi, LIN Zhi-ming, WEI Bao-quan

(East China Jiaotong Univ., Nanchang 330013, China)

Abstract: As the human-computer interface, the touch panel makes the system easily operated and handled. The thesis introduces the hardware and software implementation of the touch screen module using ADS7846 controller based on S3C4510B microprocessor development platform. And its device driver and applications for Linux embedded operating system is designed.

Key words: S3C4510B; Linux; ADS7846; touch screen; driver program